NPS55-82-033

NAVAL POSTGRADUATE SCHOOL

Monterey, California

GP-GN:

AN APPROACH CERTAIN TO LARGE-SCALE, MULTIOBJECTIVE INTEGER PROGRAMMING MODELS

by

James P. Ignizio

December 1982

Approved for public release; distribution unlimited.

Prepared for:

Naval Postgraduate School
Monterey, Ca 93940

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

Rear Admiral J. J. Ekelund
Superintendent

David A. Schrady
Provost

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS55-82-033	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) GP-GN: AN APPROACH CERTAIN TO LARGE-SCALE, MULTIOBJECTIVE INTEGER PROGRAMMING MODELS		5. TYPE OF REPORT & PERIOD COVERED Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) James P. Ignizio		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, Ca 93940		12. REPORT DATE December 1982
		13. NUMBER OF PAGES 34
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Generalized Networks
Goal Programming

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A large number of real world problems may be characterized via a multiobjective integer mathematical programming model. However, the solution to truly large-scale problems of such a type has been a difficult task. In this paper, we present a hybrid approach, combining generalized goal programming and generalized networks, for the modeling of such problems. Once such a model has been developed, it may then be possible to employ the solution procedures of generalized networks to efficiently obtain a solution - particularly if the resultant hybrid model is, fundamentally, a multiobjective generalized network.

DD FORM 1473 1 JAN 73 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ABSTRACT

A large number of real world problems may be characterized via a multiobjective integer mathematical programming model. However, the solution to truly large-scale problems of such a type has been a difficult task. In this paper, we present a hybrid approach, combining generalized goal programming and generalized networks, for the modeling of such problems. Once such a model has been developed, it may then be possible to employ the solution procedures of generalized networks to efficiently obtain a solution - particularly if the resultant hybrid model is, fundamentally, a multiobjective generalized network.

1. INTRODUCTION

Goal programming (GP) or, more appropriately generalized goal programming (GGP) [21, 24, 27], is one of several approaches that have been proposed for the modeling, solution and analysis of the multiobjective mathematical programming problem. However, GP has distinguished itself from most alternate multiobjective methods in its computational efficiency and evidence of widespread, actual implementation. Even the most adament of its critics recognizes that GP is, as of now, the "workhorse" of the multiobjective mathematical programming methods.

The computational efficiency of GP is particularly evident in dealing with either linear or nonlinear multiobjective models having continuous variables [21, 24, 27, 28, 29]. Here, GP has shown equivalent performance to that available in conventional (i.e., single objective) algorithms. However, when dealing with models which involve integer-valued variables, the performance of various (exact) integer GP algorithms has been less than impressive. As a result, the majority of truly large-scale integer GP problems have been solved by various heuristic approaches [20, 21, 23, 30].

Relatively recently, there have been some substantial accomplishments in solving single-objective integer programming problems via generalized networks. For some problems, such an approach has been shown to be up to several hundred times faster (and with less round-off error and reduced storage requirements) than available through conventional software [6, 7, 8, 12, 13, 14, 19, 37, 39]. One deficiency of such approaches, at least in the opinion of the multiobjective advocate, has been the focus on a single objective.

Purpose

The purpose of this paper is to report on an ongoing research effort that is directed toward the development of a hybrid approach which combines GP and Generalized Networks (GN). As a result, we term the hybrid approach "GP-GN". A few previous papers [22, 26, 36, 38] have addressed earlier results in this area and thus, herein, we expand on this topic with our emphasis focused primarily on the modeling aspects of GP-GN since there is an abundance of literature readily available in regard to the algorithms of generalized networks [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 31, 35, 37, 39, 40].

2. TWO GP MODELS

There are a wide variety of GP models including weighted GP, lexicographic GP (i.e., the use of the so-called "preemptive priority" concept), minmax GP (which includes fuzzy programming), and interactive GP (which is used to generate a subset of nondominated solutions) [24, 25, 27, 30, 34]. However, in this paper we shall initially restrict our attention to two particular forms of GP: (1) weighted GP and (2) lexicographic GP. The mathematical models used to represent these two types of GP are then presented below.

Weighted GP Model

Here, we wish to find $\bar{x}$ so as to:

$$\text{minimize } z = \sum_{i \varepsilon G} (u_i n_i + w_i \rho_i) \tag{2.1}$$

$$f_i(\bar{x}) \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} b_i \qquad i \varepsilon F \tag{2.2}$$

$$f_i(\bar{x}) + n_i - \rho_i = b_i \qquad i \varepsilon G \tag{2.3}$$

$$\bar{x}, \bar{n}, \bar{\rho} \geq \bar{0} \tag{2.4}$$

where:

F = the set of <u>rigid</u> constraints

G = the set of goals (where the right-hand side value represents a target value, or aspiration level, for the specific goal)

n_i = the negative deviation of goal i

ρ_i = the positive deviation of goal i

u_i, w_i = the weights associated with n_i and ρ_i respectively

Lexicographic GP Model

If the problem goals are rank ordered according to importance, we can model the problem as follows. Find $\bar{x}$ so as to determine the lexicographic minimum of:

$$\bar{a} = \{a_1, \ldots, a_k, \ldots, a_K\} \tag{2.5}$$

s.t.

$$f_i(\bar{x}) + \eta_i - \rho_i = b_i \qquad \forall i \tag{2.6}$$

$$\bar{x}, \bar{\eta}, \bar{\rho} \geq \bar{0} \tag{2.7}$$

where:

$\bar{a}$ = the "achievement vector" for which we seek the lexicographic minimal value.

a_k = a function of the goal deviation variables (η, ρ) that are to be minimized at priority level k. That is:

$$a_k = g_k(\bar{\eta}, \bar{\rho}) \tag{2.8}$$

Solving the Models

If the decision variables $(\bar{x})$ are permitted to take on fractional values, computationally efficient algorithms exist that may be used to develop the solution. For example, if the models involve strictly linear functions (i.e., all $f_i(\bar{x})$ are linear) then algorithms have been developed for use on IBM, UNIVAC and CDC computers (as well as others) that solve problems of the same size (and with approximately the same efficiency) as available through the use of conventional simplex codes. For example, the solution of lexicographic linear GP models with up to 16,000 rows (i.e., goals and rigid constraints) by as many variables as may be stored is available on IBM 360 or 370 computers having the MPSX software package [24, 27, 29].

In the case of nonlinear (weighted or lexicographic) GP models (again, with continuous variables), computer codes have been developed (and applied to a variety of real world applications [21, 24, 28]) that will solve problems with several hundred rows by thousands of variables. One actual implementation (1978 [28]) involved the solution of a problem with 400 rows and 10,000 variables and was solved in approximately nine minutes on the CDC-7600 computer.

However, as mentioned, the efficiency of computer codes for GP models involving integer variables is not particularly impressive. To the author's knowledge, most truly large-scale integer GP problems actually dealt with have been solved by means of heuristic techniques. Further, such heuristics have typically been carefully tailored to handle the specific problem under consideration and, thus, an approach to the general case has been lacking.

3. INTEGER PROGRAMMING AND NETFORM

There are numerous approaches to the modeling (and solution) of generalized networks. For purposes of illustration only, we will employ models and notation used in one of these approaches, known as NETFORM [6, 7, 12, 14]. In this section, we shall briefly summarize the NETFORM modeling approach to conventional (i.e., single objective) problems.

The NETFORM Arc

The building blocks of the NETFORM model are its nodes - and the arcs connecting these nodes. Figure 3-1 depicts a general NETFORM arc and its associated nodes. The origin node is node "i" and the terminal node is node "j". The (directed) arc, "i-j", connects these nodes. The "cost" per unit flow is given as $c_{i,j}$ and is shown within the rectangle.

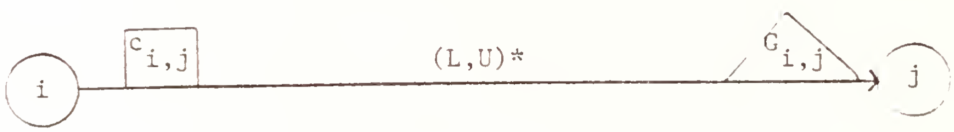

Figure 3-1: NETFORM Arc

The flow, $x_{i,j}$, has a lower bound (L) and upper bound (U) as indicated in the parentheses. Note that an asterisk will appear only if the flow across the arc must be integer valued. A particularly important, and powerful, feature of the model is given in the triangle as $G_{i,j}$. $G_{i,j}$ is, in turn, the gain factor which reflects the amplification or attenuation of the flow through the arc.

Once a problem has been so modeled, various software is available for its solution and, as mentioned, the results can be extremely impressive. One application, for the U. S. Treasury, involved a model with 5,000 nodes and 625,000 arcs. The problem was reportedly solved in 9 minutes on a UNIVAC 1108 at a cost of $90. A good commercial LP code would have, under best conditions, required about 20 hours of CPU time, at a cost of about $24,000 [14].

The Conventional Integer Programming Model

The generalized network or NETFORM modeling approach is not restricted to just those problems which naturally take on a network representation (e.g., assignment, transportation, transshipment). We can, for example, transform any single objective (and, in the next section, multiobjective) integer programming problem into an equivalent GN model. This is best illustrated via a numerical example. Consider the following model in which all variables must take on 0 or 1 values:

$$\text{minimize} \quad 3x_1 + 7x_2 + x_3$$

s.t.

$$\text{(I)} \quad x_1 + x_2 + x_3 \geq 2$$

$$\text{(II)} \quad -2x_1 + 5x_2 \geq 3$$

$$x_j \in \{0,1\} \quad j=1,2,3$$

The equivalent GN model will contain one "objective" node and this will serve as the source node for the network. Also, there will be a node, <u>for each variable</u>, connected directly to the objective (or source) node. Finally, there will be a node <u>for each constraint</u> and these constraint nodes will be directly connected to those variable nodes as associated with the variables which appear in that constraint. This leads to the GN model shown in Figure 3-2.

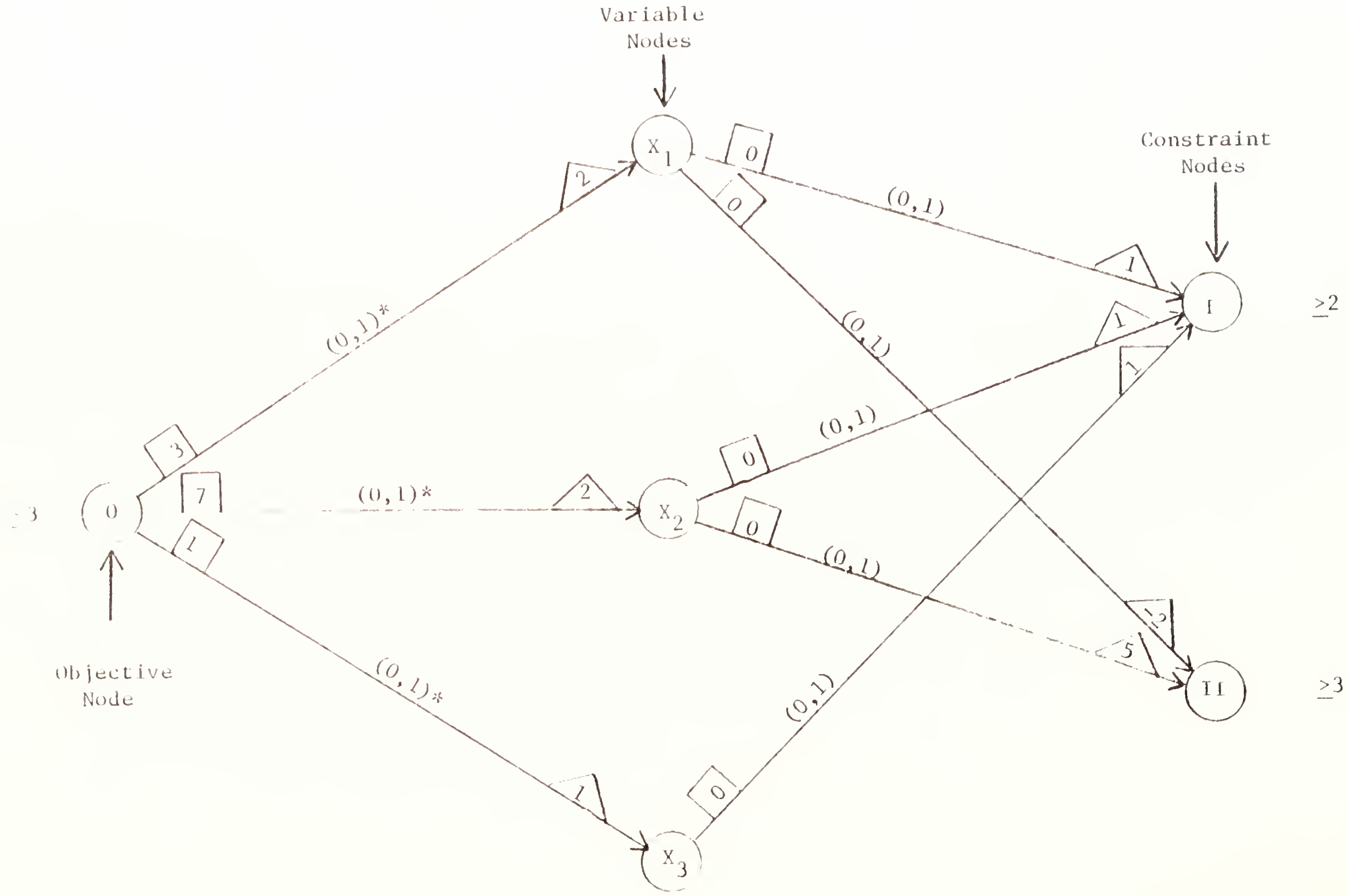

Figure 3-2; IP-GN Model

The "availability" of the objective, or source, node is shown as ≤ 3 meaning that up to three variables may take on values of one. The requirements at the terminal, or constraint, nodes are those associated with each constraint (see original model). The cost of variable 1 in the objective is 3 units as shown on the arc from the objective node to the variable one (x_1) node. The "gain" across this arc is 2 units which means that variable one appears in exactly two constraints. The arcs from the variable nodes to their associated constraint nodes have a gain factor equal to their technological coefficient in that constraint. Solving the GN model of Figure 3-2 will then provide the solution to the original mathematical model.

4. WEIGHTED GP-GN

We shall first address the modeling of the weighted integer GP model via GN. The resulting concepts and approach may then be extended so as to encompass the subject of the following section: lexicographic integer GP.

The easiest way to explain the GP-GN approach is via a simple, illustrative example. Consider the following weighted integer GP model:

· Find $\bar{x}$ so as to

$$\text{minimize} \quad z = 5\eta_1 + 2\rho_2$$

$$\begin{array}{llr}
\text{(I)} & x_1 + x_2 + x_3 & \leq 2 \\
\text{(II)} & 3x_1 \qquad\quad + 2x_3 + \eta_1 - \rho_1 & = 4 \\
\text{(III)} & 3x_1 + 2x_2 + 7x_3 + \eta_2 - \rho_2 & = 5 \\
 & x_j \in \{0,1\} \qquad\qquad \forall j &
\end{array}$$

Approach One

Recalling the modeling approach used in the conventional integer programming model, as illustrated in the previous section, we note that it may be directly applied to our example since it too is in the form of a single objective integer model. This approach would lead to a GP-GN model as shown in Figure 4-1. Here, we simply treated the goal deviation variables (η_1, η_2, ρ_1 and ρ_2) as model variables and added the corresponding variable nodes.

We choose, however, not to use this approach but, rather, employ an equivalent alternate modeling scheme, in the construction of the weighted integer GP-GN model. We discuss this technique below.

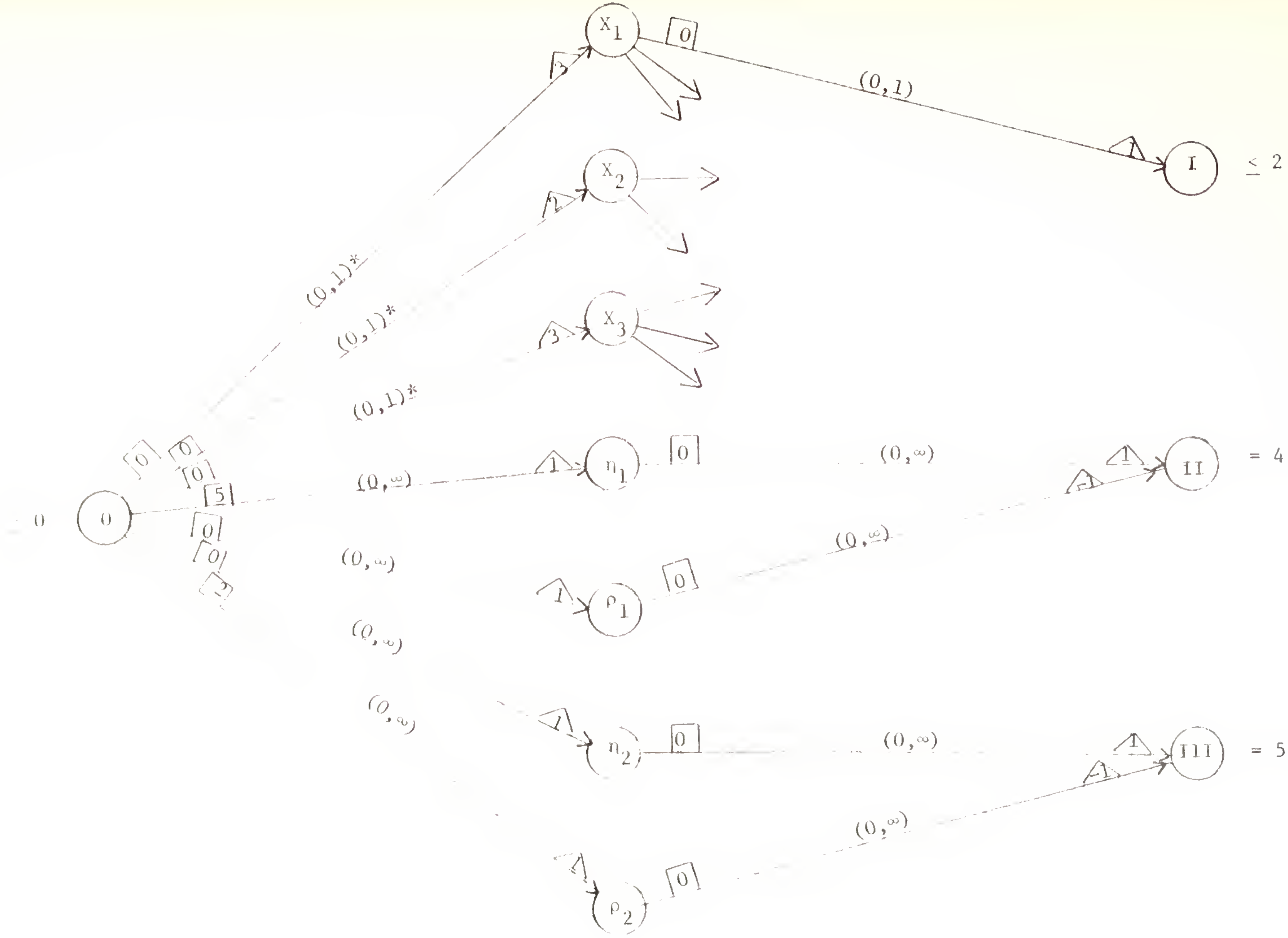

Figure 4-1: GP-GN Model: Approach One

Approach Two

This second approach involves the incorporation of a new, "complex" node. This special node, denoted as the goal node, is composed of a simple terminal node plus two additional arcs. A general goal-node is shown in Figure 4-2. In this figure, we note that node A is, in essence, a "soft" constraint node. That is, it allows for some deviation either above or below the goal target value, b_i.

The input to node A reflects the actual value achieved for that goal, which we may call a_i. The desired value, b_i, is shown to the right and, in the network, is a sink node requirement. However, if $a_i > b_i$ then there is positive deviation (or "overflow") which flows out of node A into the top, positive deviation (ρ_i) branch at a cost of w_i units per unit of flow. On the other hand, if $a_i < b_i$, then there is negative deviation (or "underflow") which requires a flow into node A - as depicted by the lower branch (n_i). The flow on this branch incurs a cost of u_i units per unit of flow. Such a goal-node will then be associated with each goal in the GP model under consideration. Combining the GN modeling process with the goal-node concept, we may model our example as shown in Figure 4-3. (Recall that numbers in rectangles represent coefficients or, in this case, weights while these in triangles refer to the gain across the arc.)

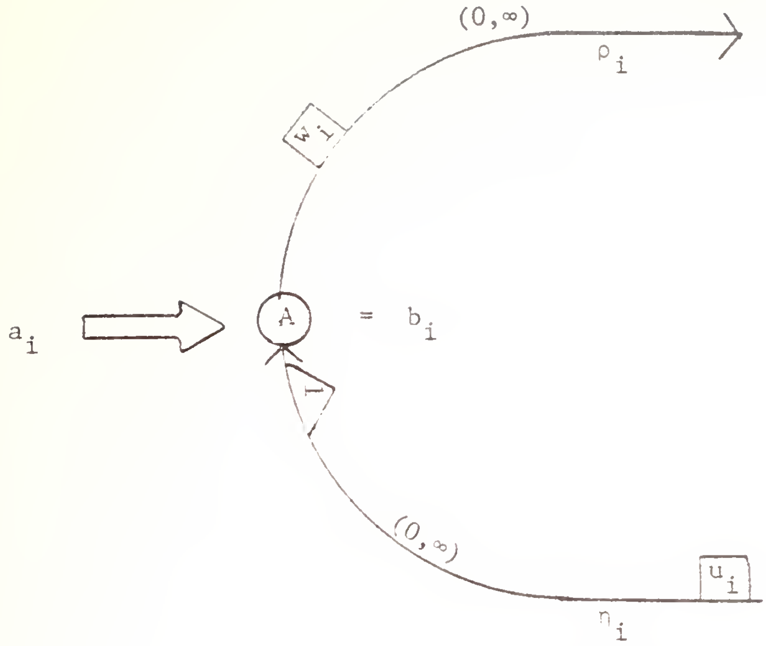

Figure 4-2: The Goal-Node

To summarize, the use of the goal-node leads to a weighted integer GP model wherein:

(1) There is a single source node (node 0) with a "supply" equal to the number of zero-one variables in the problem.

(2) There are arcs from the source node directly to a set of variable nodes (i.e., decision variables only). The multiplier (gain) on the arc indicates the number of constraints <u>plus</u> goals in which the associated variables appears.

(3) The variable nodes are connected, via arcs, to the constraint and goal-nodes in which they appear. The multiplier for these arcs is equal to the technological coefficient of the variable in the respective constraint or goal.

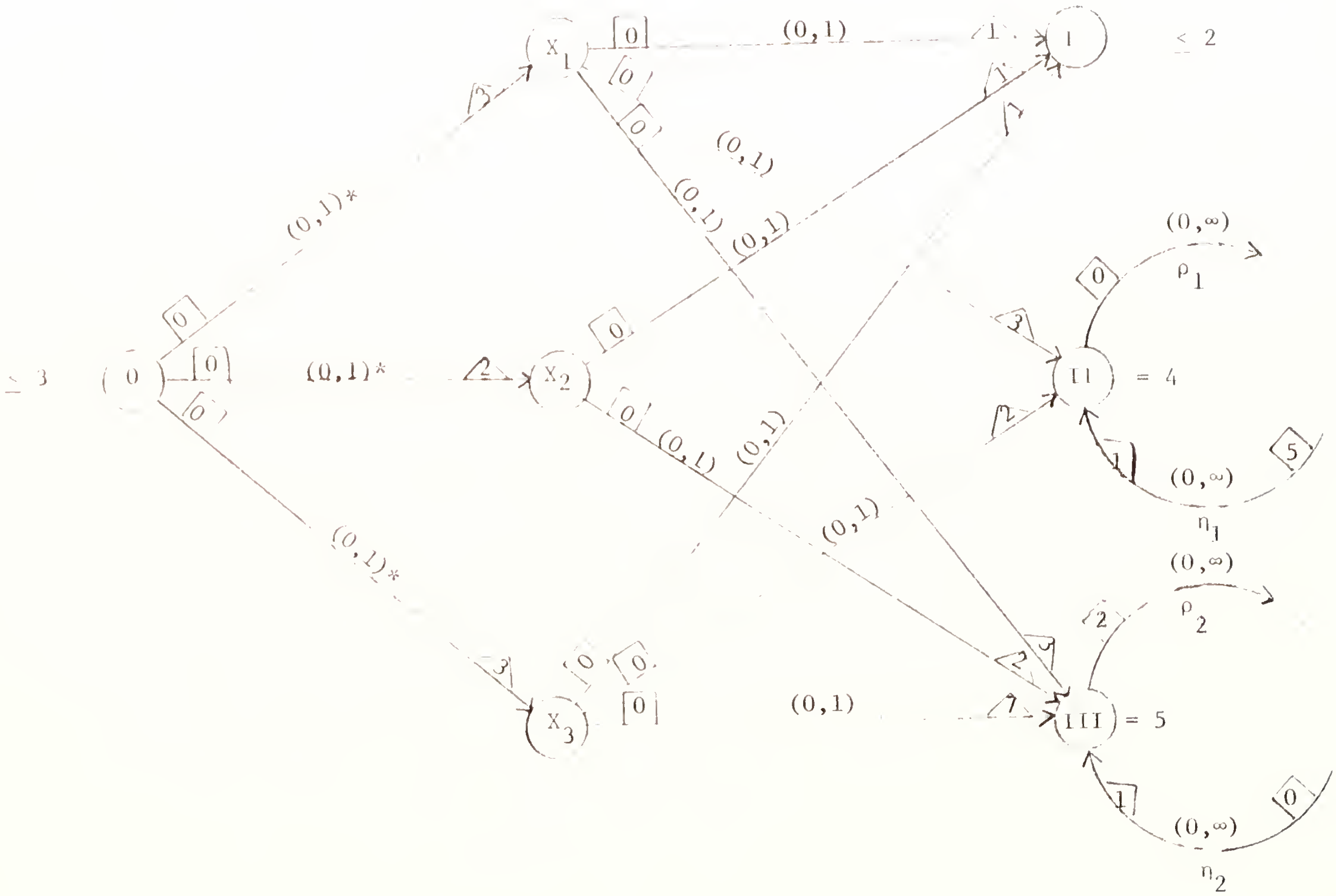

Figure 4-3: GP-GN Model: Approach Two

(4) A constraint node (see node I) is then a simple, terminating node. The "demand" of that node is shown to its right. A goal-node is also a terminating node but is, as noted, a complex node (as shown for goals II and III).

When the standard assumptions (see the references) of generalized networks are satisfied, solution to this form of the weighted integer GP-GN model is available via the appropriate generalized network software. The solution thus found, for our example, would be:

$$\bar{x}^* = (1,0,0) \text{ or } (1,1,0)$$

and $z^* = 5$

We now proceed to the second form of the GP model, i.e., lexicographic GP.

5. LEXICOGRAPHIC GP-GN

The development of the network model for lexicographic integer GP problems follows the same basic procedure as outlined, in the previous section, for weighted integer GP. However, we shall form a <u>sequence</u> of weighted integer GP-GN models for the representation of the lexicographic model where the total number of models in each sequence shall be, at worst, equal to the number of priority levels in the original lexicographic GP mathematical model.

We shall denote this process as Sequential GP-GN, or SGN. SGN, in turn, is simply the network equivalent of the approach known as SGP (i.e., sequential goal programming) that has been employed for approximately 15 years in modeling and solving large-scale GP models [24, 27, 29]. The procedure may be described as follows. First, we formulate that portion of the lexicographic GP model associated with priority one. This submodel is simply a weighted integer GP model and thus may be solved directly via the weighted GP-GN approach as described previously. Next, we form the GP model associated with priority one <u>and</u> two. However, we augment this model with the additional condition (i.e., a new rigid constraint) which restricts the next solution derived to one which does not degrade the solution found for the previous submodels [i.e., $g_1(\bar{\eta},\bar{\rho}) = a_1^*$]. This process is continued until all priority levels (and associated submodels) have been solved. The last solution is the solution to the original lexicographic GP model. The process may be summarized via the algorithm provided below.

SGN Algorithm

Step 1: Develop the lexicographic integer GP model as shown in (2.5) through (2.8). Set k = 1.

Step 2: Establish the weighted integer GP model associated with priority k (i.e., priority one). This is:

$$\text{minimize } a_1 = g_1\ (\bar{n},\bar{p})$$

s.t.

$$f_i(\bar{x}) + n_i - p_i = b_i \qquad i \in P_1$$

$$\bar{n},\ \bar{p} \geq \bar{0}$$

$$x_j \in \{0,1\} \qquad \forall j$$

Step 3: Convert the present GP submodel to the equivalent weighted integer GP-GN model and solve, via the appropriate GN software, for the optimal value of a_k (i.e., a_k*).

Step 4: If k = K, stop as the last solution ($\bar{x}$*) is the solution to the original lexicographic GP model. If k < K, go to Step 5.

Step 5: Set k = k + 1 and establish the next (weighted) GP submodel in the sequence where this submodel is given by:

$$\text{minimize } a_k = g_k\ (\bar{n},\bar{p})$$

s.t.

$$f_i\ (\bar{x}) + n_i - p_i = b_i \qquad i \in T_k$$

$$g_t\ (\bar{n},\bar{p}) = a_t^* \qquad t=1,2,\ldots,k-1$$

$$\bar{n},\bar{p} \geq \bar{0}$$

$$x_j \in \{0,1\} \qquad \forall j$$

and T_k is the set of all goals and constraints in priority levels 1 through k.

Step 6: Repeat Steps 3 through 5 until the entire model has been solved. The final value of $\overline{x}^*$ is the optimal program for the original lexicographic integer GP model.

Implementation

Implementation of the SGN process would be achieved via the use of a buffer program in conjunction with the GN software. The buffer program simply automates the above process by forming each submodel, delivering the submodel to the GN software, and forming the necessary augmented constraint associated with the intermediate GN outputs. In actual practice, however, there are a number of refinements that may be added so as to substantially improve overall computational efficiency. We touch on just a few of these in the example to follow as well as in the next section.

Example

In order to simply illustrate the basic procedure, we shall formulate the sequence of models associated with the problem shown below:

Find $\overline{x}$ to lexicographically

minimize:

$$\overline{a} = \{(\rho_1),\ (5\eta_2 + 2\rho_3),\ (\eta_3)\} \tag{5.1}$$

s.t

$$x_1 + x_2 + x_3 + \eta_1 - \rho_1 = 2 \tag{5.2}$$

$$3x_1 + 2x_3 + \eta_2 - \rho_2 = 4 \tag{5.3}$$

$$3x_1 + 2x_2 + 7x_3 + \eta_3 - \rho_3 = 5 \tag{5.4}$$

$$x_j \in \{0,1\} \text{ and } \overline{\eta},\ \overline{\rho} \geq \overline{0} \tag{5.5}$$

The three models developed to support the SGN process are shown in Figures 5-1 through 5-3 and we shall briefly comment on the construction of each.

In Figure 5-1, the GP-GN model for priority level one is shown. The solution gives $a_1^* = 0 = \rho_1$ and we move to the model for priority levels one and two, as shown in Figure 5-2. However, we may simplify constraint node I at this time. That is, since ρ_1 must be kept to a value of zero, equation (5.2) is equivalent to:

$$x_1 + x_2 + x_3 + \eta_1 = 2$$

or, more simply,

$$x_1 + x_2 + x_3 \leq 2$$

As a result, the goal node for goal I (as in Figure 5-1) may be replaced by the simpler constraint node shown in Figure 5-2.

Solving the GP-GN model of Figure 5-2 gives:

$$a_2^* = 5 = 5\eta_2 + 2\rho_3 \tag{5.6}$$

which is the augmented constraint that should be added to the next GP-GN model. However, as indicated in Figure 5-3, relationship (5.6) can be included in the network by simply modifying the two goal nodes as shown. That is, we add the augmented source node "A" with an input of exactly 5 units (i.e., the value of a_2^*). The two arcs emanating from node A then assure us that, in any solution to the GP-GN network of Figure 5-3, relationship (5.6) must be satisfied.

Solving the final GP-GN model of Figure 5-3 gives us:

$$a_3^* = 0$$

$$\bar{x}^* = (1,1,0)$$

and thus the solution to the example of (5.1) through (5.5) is:

$$\bar{a}^* = (0,5,0)$$

$$\bar{x}^* = (1,1,0)$$

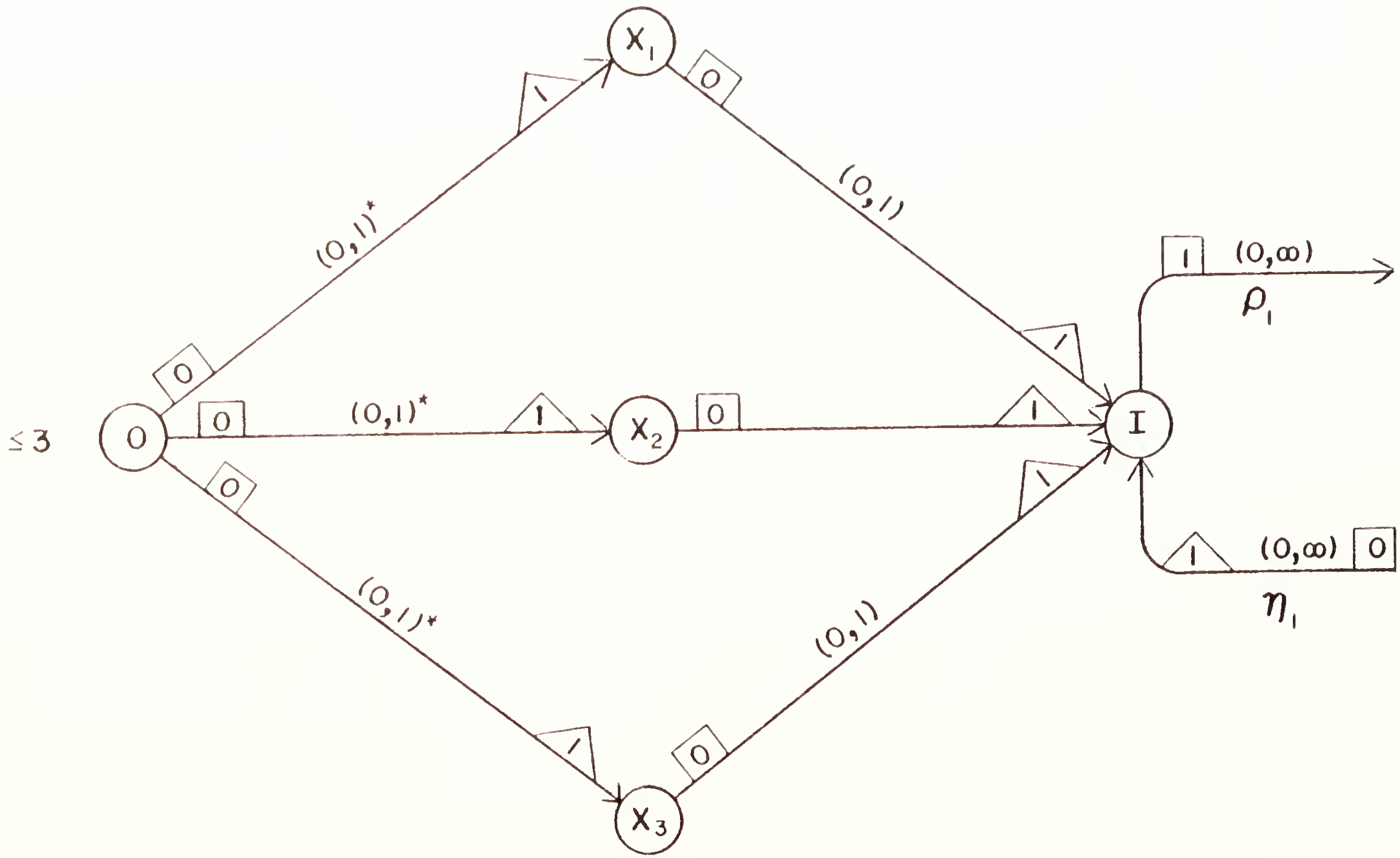

Figure 5-1: Priority One ($a_1* = 0 = \rho_1$)

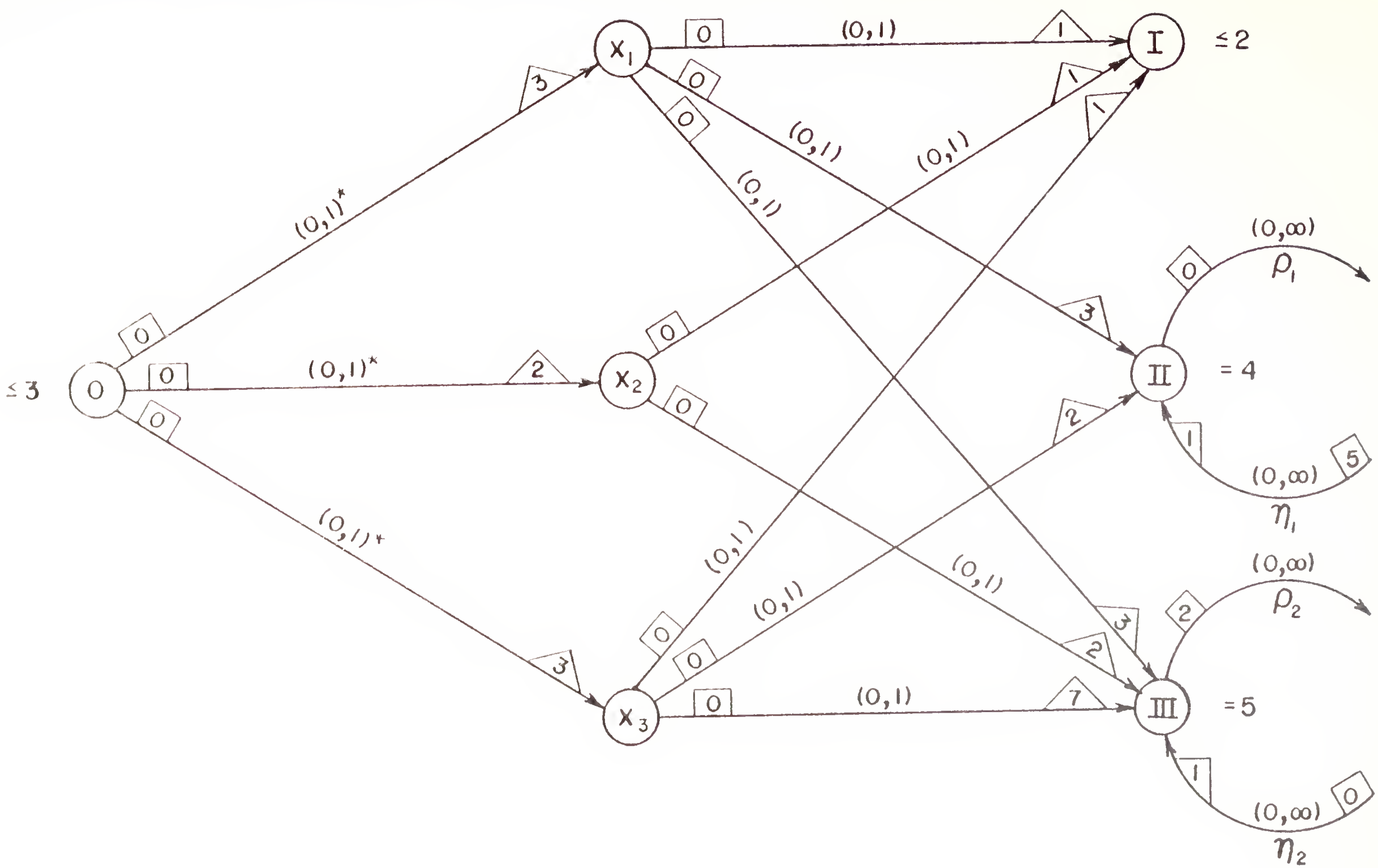

Figure 5-2: Priority Two ($a_2^* = 5 = 5\eta_2 + 2\rho_3$)

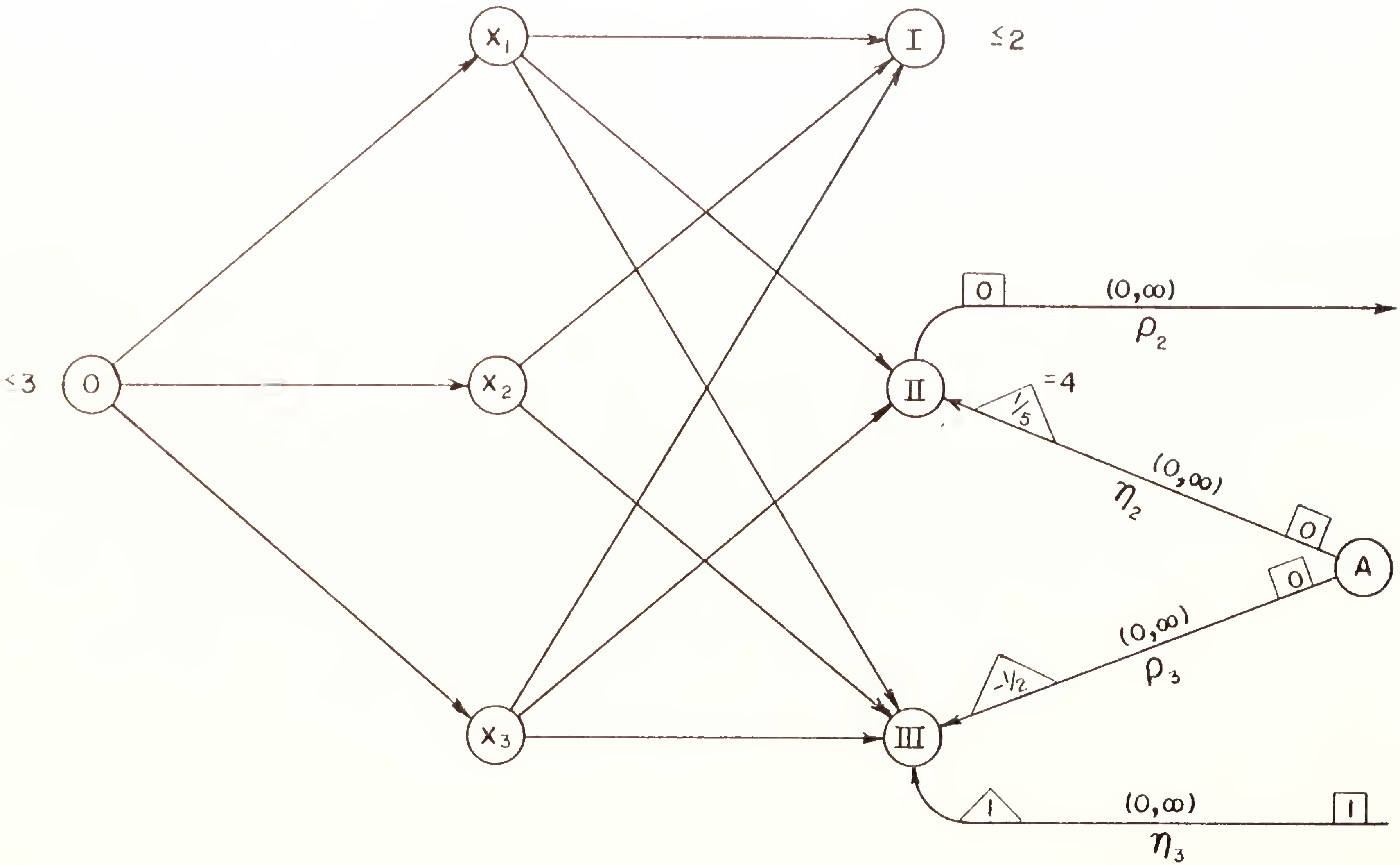

Figure 5-3: Priority Three ($a_3^* = 0 = \rho_3$)

6. REFINEMENTS: LEXICOGRAPHIC GP-GN

Just as there have been found to be numerous refinements to SGP (such as "column drop", "early stop", etc. [27]), there are similar refinements possible for SGN. In fact, the very use of the network representation provides for a number of straightforward, but potentially powerful, refinements that may greatly enhance computational efficiency. We discuss a few of the more obvious of these below.

Augmented Constraint

Step 5 of the SGN algorithm indicates the addition, for each submodel except the first, of the augmented constraint:

$$g_t\ (\bar{\eta},\bar{\rho}) = a_t^* \tag{6.1}$$

From this, it would appear that we are increasing the complexity of the network equivalent when, in fact, quite often just the opposite occurs.

To illustrate, consider first the case in which the augmented constraint (6.1) contains deviation variables from more than one goal (i.e., there exist several goals at the previous priority level). This case may be considered via the use of the specific example given below:

$$g_t(\bar{\eta},\bar{\rho}) = w_1\rho_1 + u_2\eta_2 = a_t^* \tag{6.2}$$

In this example, we have two goals at priority t. The goal-nodes for these two goals are given, originally, as shown in Figure 6-1. Note that, if the achieved level of (6.2) is zero (i.e., $a_t^* = 0$), a simplification is possible. That is, if $a_t^* = 0$ then (6.2) becomes:

$$w_1\rho_1 + u_2\eta_2 = 0$$

which can only be satisfied when there is no flow across either the ρ_1 or η_2 arc. For this case, we simply remove these two arcs from their goal-nodes of Figure 6-1 in all further calculations.

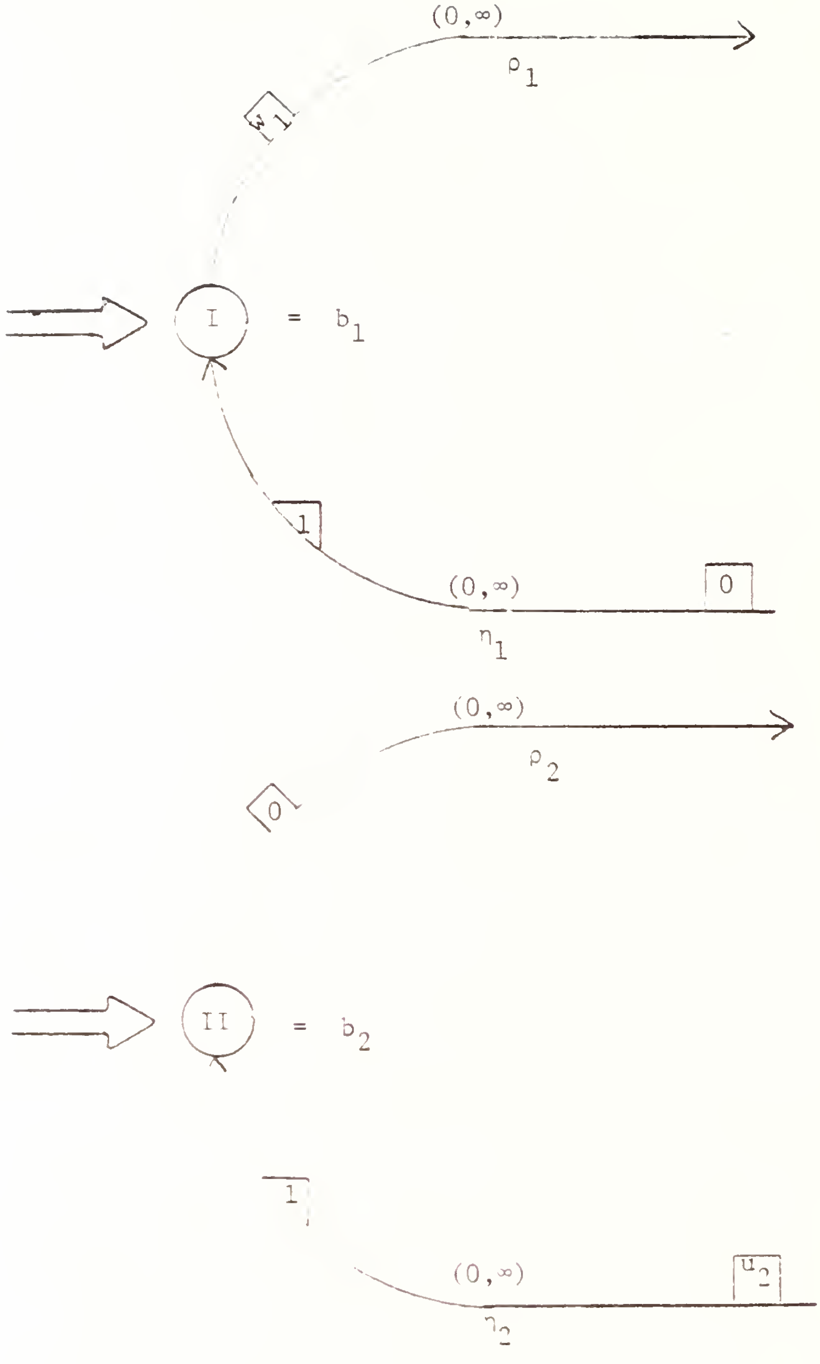

Figure 6-1: $g_t\ (\bar{\eta},\bar{\rho}) = w_1\rho_1 + u_2\eta_2$

When the optimal solution to (6.2) is non-zero (i.e., $a_t^* > 0$), we may use the network modification already illustrated in Figure 5-3, for our example of section 5.

Next, let us consider the case wherein, for the priority level previously solved, there is but <u>one</u> associated goal. That is, the associated augmented constraint is of the general form shown below:

$$g_t(\bar{\eta},\bar{\rho}) = w_i\,\rho_i + u_i\,\eta_i = a_t^* \tag{6.3}$$

Now, if the value of a_t^* is zero, we have an obvious simplification designated as the "collapse" of the associated goal node. This is illustrated, for a specific form of (6.3), in our previous example of section 5 wherein the goal node of Figure 5-1 "collapsed" into a standard constraint node in Figure 5-2.

Variable Reduction

It may also be possible, when performing the SGN algorithm, to actually eliminate one or more variables (and thus their associated nodes and arcs) from the model. The most obvious case is when a constraint node (either an original constraint node or a collapsed goal-node) can only be satisfied by one specific combination of those variables whose arcs lead to that node.

Early Stop

If, through variable elimination (as directed above), <u>all</u> the variable values are fixed, the problem is solved and thus it is not necessary to consider further submodels in the sequence.

Nonlinear Functions

The processes previously described are also applicable in principle, to nonlinear integer GP models. Here, the single arc from a variable node to a constraint or goal-node would be replaced by several arcs. Each of these arcs would represent a linear approximation, over a given interval, of the nonlinear function.* The interval, in turn, is controlled via the upper and lower bounds on flow across each of the multiple arcs associated with the specific variable-constraint/goal combination.

Minmax GP and Fuzzy Programming

In Section 2, we presented only two (i.e., weighted GP and lexicographic GP) forms of GP. Another form of GP (see the references for further details [27, 32, 41]) is that known as minmax GP. In minmax GP, we strive to minimize the worst single goal deviation. Two natural and straightforward extensions of minmax GP are: lexicographic minmax GP and an approach known as "fuzzy programming" [27, 32, 41]. By means of approaches similar to those discussed in this paper, these models may also be represented and solved via the GN approach and, consequently, a detailed discussion is believed unnecessary.

Augmented GP

Yet a further class of GP is that known as augmented GP [27, 34]. Augmented GP may be used to derive (via minimal interaction with the decision maker) a subset of all the nondominated solutions which exist for a given problem. Such an approach is implemented by means of extensions to the GP-GN concept and is the topic of a present research effort.

---

*The restrictions typical of all such approximations must, of course, hold.

7. SUMMARY

In this paper we have provided and described, via example, the results of an on-going research effort dedicated to the modeling, solution and analysis of multiobjective integer programming problems. This has been accomplished by means of combining generalized goal programming [21, 24, 25, 27, 28, 29, 30, 34] with generalized networks [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 22, 26, 31, 33, 35, 37, 38, 39, 40]. The resulting methodology, denoted as GP-GN may be applied to any multiple objective model which conforms to the standard assumptions of generalized networks. For such problems, the GP-GN approach should provide for a computationally efficient method of problem solving.

8. ACKNOWLEDGEMENTS

A portion of the support for this research was provided via a National Research Council (NRC) Senior Research Award. The sponsoring institute for this award was the Naval Postgraduate School in Monterey, California.

8. REFERENCES

1. Bradley, G. "Survey of Deterministic Networks," AIIE Trans., Vol. 7, No. 3, 1975, pp. 222-234.

2. Charnes, A., D. Karney, D. Klingman, and J. Stutz. "Past, Present and Future of Large Scale Transshipment Computer Codes and Applications," Computers & Ops. Res., Vol. 2, 1975, pp. 71-81.

3. Cunningham, W. "A Network Simplex Method," Math. Prog., Vol. II, 1976, pp. 105-116.

4. Daniel, R. C. "LP-based Mathematical Programming - The Significance of Recent Developments," J. Opl. Res., Vol. 32, 1981, pp. 113-119.

5. Elam, J., F. Glover, and D. Klingman. "A Strongly Convergent Primal Simplex Algorithm for Generalized Networks," Math. Ops. Res., Vol. 4, No. 1, 1979, pp. 39-59.

6. Glover, F., J. Hultz and D. Klingman. "Improved Computer-Based Planning Techniques, Part I," Interfaces, Vol. 8, No. 4, August 1978, pp. 16-25.

7. Glover, F., J. Hultz and D. Klingman. "Improved Computer-Based Planning Techniques, Part II," Interfaces, Vol. 9, No. 4, August 1979, pp. 12-20.

8. Glover, F., D. Karney and D. Klingman. "Implementation and Computational Comparisons of Primal, Dual, and Primal-Dual Computer Codes for Minimum Cost Network Flow Problems," Networks, Vol. 4, No. 3, 1974, pp. 191-212.

9. Glover, F. and D. Klingman. "On the Equivalence of Some Generalized Network Problems to Pure Network Problems," Math. Prog., Vol. 4, No. 3, 1973, pp. 369-378.

10. Glover, F. and D. Klingman. "A Note on Computational Simplification in Solving Generalized Transportation Problems," Transportation Science, Vol. 7, 1973, pp. 351-361.

11. Glover, F. and D. Klingman. "Capsule View of Future Developments on Large-Scale Network and Network Related Problems," Res. Rep. CCS 238, Centre for Cybernetic Studies, Univ. of Texas, 1975.

12. Glover, F. and D. Klingman. "Network Application in Industry and Government," AIIE Trans., Vol. 9, No. 4, 1977, pp. 363-376.

13. Glover, F. and D. Klingman. "The Simplex SON Algorithm for LP/embedded Network Problems," Res. Rep. CCS-317, Centre for Cybernetic Studies, Univ. of Texas, revised Jan. 1979.

14. Glover, F., D. Klingman and C. McMillan. "The NETFORM Concept," Proc. of ACM, Seattle, 1977.

15. Glover, F. and J. Mulvey. "Equivalence of the 0-1 Integer Programming Problem to Discrete Generalized and Pure Networks," Ops. Res., Vol. 28, No. 3, Part II, May-June 1980, pp. 829-836.

16. Glover, F. and L. Woolsey. "Further Reduction of Zero-One Polynomial Programming Problems to Zero-One Linear Programming Problems," Ops. Res., Vol. 21, No. 1, 1973, pp. 156-161.

17. Glover, F. and E. Woolsey. "Converting the 0-1 Polynomial Problem to a 0-1 Linear Program," Ops. Res., Vol. 22, No. 1, 1974, pp. 180-182.

18. Golden, B. "Shortest Path Algorithms: A Comparison," Ops. Res., Vol. 24, No. 6, 1976, pp. 1164-1168.

19. Golden, B., M. Ball and L. Bodin. "Current and Future Research Directions in Network Optimization," Comput. & Ops. Res., Vol. 8, 1981, pp. 71-81.

20. Ignizio, J. P. "A Heuristic Solution to Generalized Covering Problems," Ph.D. Dissertation, Virginia Polytechnic Institute, 1971.

21. Ignizio, J. P. Goal Programming and Extensions, Lexington, MA: D. C. Heath, 1976.

22. Ignizio, J. P. "The Goal-Node and Goal Programming via NETFORM," Working Paper, Pennsylvania State University, October 1978.

23. Ignizio, J. P. "Solving Large-Scale Problems: A Venture into a New Dimension," Jrl. Opl. Res., Vol. 31, 1980, pp. 217-225.

24. Ignizio, J. P. "An Introduction to Goal Programming: With Applications in Urban Systems," Jrl. Computers, Environment & Urban Systems, Vol. 5, 1980, pp. 15-33.

25. Ignizio, J. P. "The Determiniation of a Subset of Efficient Solutions via Goal Programming," Computers & Ops. Res., Vol. 8, No. 1, 1981, pp. 9-16.

26. Ignizio, J. P. "An Approach to the Modeling and Analysis of Multi-objective Generalized Networks," Working Paper, Pennsylvania State University, November 1981.

27. Ignizio, J. P. Linear Programming in Single & Multiple-Objective Systems, Englewood Cliffs, NJ: Prentice-Hall, 1982.

28. Ignizio, J. P. et al. "Advances in Multicriteria Engineering Design: Vols. I, II and III," U. S. Army Contract DASG60-77-C-0078, June 1979.

29. Ignizio, J. P. and J. H. Perlis. "Sequential Linear Goal Programming: Implementation via MPSX," Comput. & Ops. Res., Vol. 3, No. 1, 1980, pp. 217-225.

30. Ignizio, J. P., D. F. Palmer and C. M. Murphy. "A Multicriteria Approach to the Overall Design of Distributed Computing Systems," Special Issue on Supersystems, IEEE Trans. on Comp., March 1982.

31. Karney, D. and D. Klingman. "Implementation and Computational Study on In-Core Out-of-Core Primal Network Code," Ops. Res., Vol. 24, No. 6, 1976, pp. 1056-1077.

32. Kickert, W. J. M. Fuzzy Theories on Decision Making, Leiden: Martinus-Nijhoff, 1978.

33. Klastorin, T. D. "On the Maximal Covering Location Problem and the Generalized Assignment Problem," Management Science, Vol. 25, No. 1, Jan. 1979, pp. 107-112.

34. Masud, A. S. and C. L. Hwang. "Interactive Sequential Goal Programming," Jrl. Opl. Res., Vol. 32, 1981, pp. 391-400.

35. Maurras, J. "Optimization of the Flow Through Networks with Gains," Math. Prog., VOl. 3, 1972, pp. 135-144.

36. McGinnis, L. F. and V. V. Rao, "On Goal Programming in Networks," Proc. 13th S. E. TIMS, Clemson, SC, 1978.

37. Mulvey, J. "Testing of Large-Scale Network Optimization Program," Math. Prog., Vol. 15, No. 3, 1978, pp. 291-314.

38. Price, W. L. "Solving Goal-Programming Manpower Models Using Advanced Network Codes," J. Opl. Res., Vol. 29, No. 12, 1978, pp. 1231-1239.

39. Ross, G. T. and R. M. Soland. "A Branch and Bound Algorithm for the Generalized Assignment Problem," Math. Prog., Vol. 8, No. 1, 1975, pp. 91-104.

40. Truemper, K. "An Efficient Scaling Procedure for Gain Networks," Networks, Vol. 6, No. 2, 1976, pp. 151-159.

41. Zimmermann, H.-J. "Fuzzy Programming and Linear Programming with Several Objective Functions," Fuzzy Sets and Systems, Vol. 1, 1978, pp. 45-56.

DISTRIBUTION LIST

	NO. OF COPIES
Library, Code 0142 Naval Postgraduate School Monterey, CA 93940	4
Dean of Research Code 012A Naval Postgraduate School Monterey, CA 93940	1
Library, Code 55 Naval Postgraduate School Monterey, CA 93940	1
Professor J. P. Ignizio Code 55 Naval Postgraduate School Monterey, CA 93940	40